THE UNIVERSITY of EDINBURGH

Edinburgh Research Explorer

GPU-Accelerated Transient Electro-Magnetic Modelling

Citation for published version:
Liu, Y, Ziolkowski, A & Stoffa, P 2019, GPU-Accelerated Transient Electro-Magnetic Modelling. in *EAGE Annual 81st Conference and Exhibition.*, Tu_P05_10.

Link:
Link to publication record in Edinburgh Research Explorer

Published In:
EAGE Annual 81st Conference and Exhibition

General rights
Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.

OPEN ACCESS

Download date: 05. May. 2023

Introduction

Modelling 3-D transient electromagnetic (EM) fields is important for the interpretation of multidimensional conductivity structures. Most existing modelling codes use finite-difference methods (Commer and Newman, 2004) or finite-element methods (Um et al., 2010). Low-order approximations of spatial and temporal derivatives introduce numerical errors. Stoffa and Ziolkowski (2018) present an algorithm in which the spatial propagation uses the pseudo-spectral (PS) method, while the temporal evolution uses the rapid expansion method (REM) which solves the field by a summation of Chebyshev polynomials (Carcione 2006; Pestana and Stoffa, 2010). The results are free of dispersion and accurate to the Nyquist in both space and time.

Another concern is numerical efficiency. The Chebyshev approach is well suited for parallel computing as it does not require matrix multiplication. The computationally-intensive tasks are the 3-D fast Fourier transforms and the dot products among matrices. We recognise the excellent performance of GPU computing for this aspect, and have implemented our algorithm in a CUDA C code. The performance of the code running on a K80 GPU can be over 100 times faster than the serial C code, and 20 times faster than the multi-threaded C code with 16 processors. In addition, we show that if the model consists only of 1-D or 2-D structures, the computational effort to solve a 3-D field can be reduced by up to two orders of magnitude without loss of accuracy. The performance improvements and the flexibility to choose the solution domain should further generalize the use of REM in 1-D, 2-D and 3-D problems.

We first give a brief review of the theory of REM. We then explain the parallel implementation of the code and demonstrate the power of GPU computing. We derive the theory of solving the electric field in 1-D or 2-D model space with 3-D source consideration. Finally, we present a 2.5-D modelling example to demonstrate the usefulness of the GPU-accelerated code.

3-D solution by REM

In conductive media where the displacement current can be neglected, the electric field $\boldsymbol{E}(x,y,z,t)$ satisfies the diffusion equation

$$\partial_t \boldsymbol{E} = -{\mu_0}^{-1}\boldsymbol{\sigma}^{-1}\nabla\times\nabla\times\boldsymbol{E} - \boldsymbol{\sigma}^{-1}\partial_t \boldsymbol{J_s}, \tag{1}$$

where $\boldsymbol{E}$ is the electric field consisting of three components $\boldsymbol{E}(x,y,z,t) = (E_x, E_y, E_z)^T$ with units $\mathrm{V/m}$; μ_0 is the magnetic permeability of free space ($\mu_0 = 4\pi\times10^{-7}\mathrm{H/m}$); $\boldsymbol{\sigma}$ (S/m) is the conductivity tensor and $\boldsymbol{J_s}$ $(\mathrm{A/m^2})$ is the source current density. The solution of equation 1 by REM has been fully discussed in Stoffa and Ziolkowski (2018) and Liu and Ziolkowski (2018). Equation 1 can be written in the form $\partial_t \boldsymbol{E} = \boldsymbol{G}\,\boldsymbol{E} + \boldsymbol{s}$, where $\boldsymbol{s}$ is the source term and $\boldsymbol{G}$ is the propagation matrix containing all the spatial derivatives and the conductivities. We consider an impulsive electric dipole source. Immediately after the impulse there is no source term but there is an initial field. The relaxation of the field follows an exponential operator

$$\boldsymbol{E} = exp(\boldsymbol{G}\,t)\boldsymbol{E}_0\,, \tag{2}$$

where $\boldsymbol{E_0}$ defines the initial field. The exponential term $exp(\boldsymbol{G}\,t)\boldsymbol{E_0}$ is evaluated as a sum of Chebyshev polynomials that is convergent at the truncated term M

$$\boldsymbol{E} = \sum_{k=0}^{M} b_k\,\boldsymbol{Q}_k, \tag{3}$$

where b_k is the weighting of the kth Chebyshev term, and $\boldsymbol{Q}_k$ denotes the kth Chebyshev polynomial. The Chebyshev terms are updated by the recursion relation as

$$\boldsymbol{Q}_0 = \boldsymbol{E}_0, \tag{4}$$

$$\boldsymbol{Q}_1 = \boldsymbol{F}\boldsymbol{E}_0, \tag{5}$$

$$\boldsymbol{Q}_{k+1} = 2\boldsymbol{F}\boldsymbol{Q}_k - \boldsymbol{Q}_{k-1}, \tag{6}$$

with the numerical operator $\boldsymbol{F}$ defined as

$$\boldsymbol{F} = \frac{1}{b}\boldsymbol{G} + \boldsymbol{I}, \tag{7}$$

where b is the absolute value of the maximum eigenvalue of the propagation matrix $\boldsymbol{G}$, and $\boldsymbol{I}$ is the identity matrix. The numerical structure of the modelling code exactly follows equations 4, 5 and 6 to solve the required Chebyshev terms in order. The time-domain electric field is then retrieved by a weighted summation of the Chebyshev terms, as shown in equation 3.

Parallel implementation by GPU

The major computational burden comes from the calculation of the propagation matrix at every step

$$\boldsymbol{G}\boldsymbol{Q}_k = -{\mu_0}^{-1}\boldsymbol{\sigma}^{-1}\,\mathrm{FFT}^{-1}\left[\widetilde{\boldsymbol{D}}\cdot\mathrm{FFT}(\boldsymbol{Q}_k)\right], \tag{8}$$

where the FFT operator denotes the 3-D fast Fourier transform and the differential matrix $\widetilde{\boldsymbol{D}}$ contains all the spatial derivatives transformed into the 3-D wavenumber domain. The computational task mainly consists of two parts: 1) the dot products of the 3-D matrices and 2) the 3-D FFTs. The former is embarrassingly parallel in the space/wavenumber domain, whereas the latter is not.

To demonstrate the problem, we have implemented the algorithm as a multi-threaded C code using OpenMP. The 3-D FFTs are handled by the multi-threaded FFTW package (Frigo and Johnson, 2005). The dot product is parallelized along all the three dimensions. We test the performance of the code in a homogeneous half space ($\sigma = 3\,S/m$) with 128×256×256 nodes and simulate a 200 ms time period which requires approx. 650 Chebyshev terms. Figure 1(a) shows the performance of the code as the number of processors is increased. The run time decreases with the number of processors as expected, but the speed-up factor only reaches 4 with 16 processors, i.e. an efficiency of 0.25. The low parallel efficiency is mainly due to the calculation of FFTs and the synchronization of the threads at the barriers between FFTs and matrix dot products (there are 3 barriers for full-space and 7 for half-space modelling).

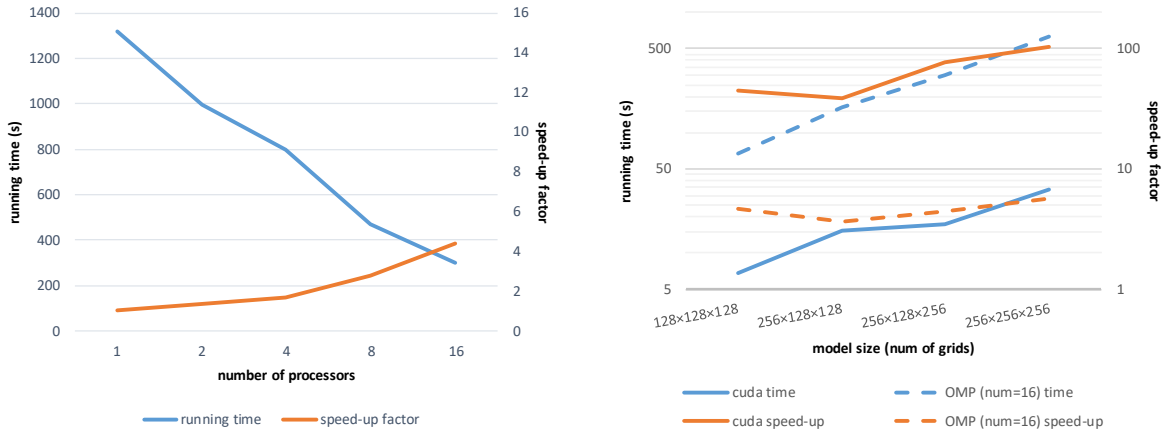

Figure 1 *(a) Run time and speed-up test with the multi-threaded C code. (b) Comparison between the GPU- and CPU-accelerated C code for different sizes of models.*

Another approach to solving this problem is to use GPU acceleration to compute the dot products and FFTs. We have implemented the algorithm using CUDA C code, to take advantages of the cuFFT algorithm when computing large-scale 3-D FFTs. The spatial grids and the nested for-loops are distributed to the GPU threads whenever possible. Data transfer between the CPU and GPU memory is time consuming. Therefore, all field components are allocated on the GPU memory and expensive transfers happen only once, no matter how many times the forward model needs to be called.

We test the GPU-accelerated code on a K80 GPU with different sizes of model and show the results in Figure 1(b). For the model size of 256×256×256 nodes, the performance of the CUDA C code is over 100 times faster than the serial code, and 20 times faster than the multi-threaded C code with 16 processors. The improvement in speed is very impressive. Furthermore, the speed-up factor increases

with the size of the model, which indicates the potential of GPU computing for large-scale problems, although the GPU memory may be a limiting factor for the moment.

Extension to 1-D/2-D model space with a 3-D dipole source

In some cases, the 3-D conductivity can be simplified to 2-D model space if one can assume the material properties are unchanged along the strike direction. The source (e.g., an electric dipole) emits a 3-D field, which yields a 2.5-D problem (e.g., Key and Ovall, 2011). The main strategy of 2.5-D EM modelling is to convert the field along the strike direction to its wavenumber domain and solve a series of 2-D equations, and then convert the field back to the spatial domain. The solution generally takes three steps: 1) convert the electric field $\boldsymbol{E}(x,y,z)$ to $\widetilde{\boldsymbol{E}}(x,k_y,z)$; 2) solve the x-z plane 2-D problem for a set of k_y values logarithmically spaced covering the required wavenumber range; and 3) interpolate the values of $\widetilde{\boldsymbol{E}}(x,k_y,z)$ for the rest of the wavenumbers and convert the field back to $\boldsymbol{E}(x,y,z)$. The 2.5-D model can easily be one order of magnitude faster than the 3-D run, as usually around ~30 wavenumbers are sufficient whereas in a 3-D problem hundreds of spatial nodes are required.

Modelling by REM is flexible between the space and wavenumber domains, as the evolution of the field in the Chebyshev domain is independent of the domain. The updating of the terms (equations 4-6) can be carried out in the domains of (k_x,k_y,k_z) or (k_x,k_y,z) or (x,k_y,z) or (x,y,z), to simulate the field in a homogeneous, 1-D layered, 2-D, or 3-D model, respectively, with the propagation matrix $\boldsymbol{G}$ modified accordingly. To incorporate the 3-D dipole, the initial field needs to be solved analytically in the desired modelling domain. By taking the Fourier transform of the space-domain initial field (Stoffa and Ziolkowski, 2018), we solve the analytical expression for the electric field excited by an impulsive dipole in the (x,k_y,z) domain as:

$$E_x(x,k_y,z,t)=\frac{Ids}{4\pi\sigma t}\exp(-\theta^2(x^2+z^2)-\frac{{k_y}^2}{4\theta^2})(-4\theta^4z^2+2\theta^2+{k_y}^2), \tag{9}$$

$$E_y(x,k_y,z,t)=-i\frac{Ids\,\theta^2}{2\pi\sigma t}\exp(-\theta^2(x^2+z^2)-\frac{{k_y}^2}{4\theta^2})xk_y, \tag{10}$$

$$E_z(x,k_y,z,t)=\frac{Ids\,\theta^4}{\pi\sigma t}\exp\left(-\theta^2(x^2+z^2)-\frac{{k_y}^2}{4\theta^2}\right)xz, \tag{11}$$

and therefore following equations 4-6 the 2.5-D EM modelling problem has been solved by REM. Furthermore, if the model is considered to be 1-D, the initial field is solved as

$$E_x(k_x,k_y,z,t)=\frac{Ids}{4\pi^{1/2}\sigma t\theta}\exp(-\frac{{k_x}^2}{4\theta^2}-\frac{{k_y}^2}{4\theta^2}-\theta^2z^2)(-4\theta^4z^2+2\theta^2+{k_y}^2), \tag{12}$$

$$E_y(k_x,k_y,z,t)=-\frac{Ids}{4\pi^{1/2}\sigma t\theta}\exp(-\frac{{k_x}^2}{4\theta^2}-\frac{{k_y}^2}{4\theta^2}-\theta^2z^2)k_xk_y, \tag{13}$$

$$E_z(k_x,k_y,z,t)=-i\frac{Ids\,\theta}{2\pi^{\frac{1}{2}}\sigma t}\exp\left(-\frac{{k_x}^2}{4\theta^2}-\frac{{k_y}^2}{4\theta^2}-\theta^2z^2\right)k_xz, \tag{14}$$

and following equations 4-6, the time-domain electric field excited by a 3-D dipole source is solved in a 1-D model, which gives another order of magnitude improvement in numerical efficiency.

A 2.5-D Example

We test the 2.5-D version of the CUDA C code with the same half-space problem as shown in Liu and Ziolkowski (2018). The 2-D model consists of 128×128 nodes with 10 m spacing along the x- and z-direction, and 30 k_y wavenumbers logarithmically spaced from 10^{-4} to the Nyquist wavenumber. The model consists of a 300 m water layer (σ=3 S/m), a VTI anisotropic sediment layer (σ_h=1 S/m; σ_n=0.5 S/m) beneath the water and a 200m×100m resistive anomaly (σ =0.02 S/m) inserted 100 m below the centre of the seabed. The impulsive dipole source is located 150 m below the air-earth interface in the centre and the receivers are located on the sea bed.

Figure 2 compares the inline time-domain response along the receiver line between the 2.5-D and 3-D modelling results. The accuracy of 3-D REM has been demonstrated by Liu and Ziolkowski (2018). The 2.5-D results show excellent agreement at early-, mid- and late-times with the relative error all below 1%, which demonstrates the maintenance of the accuracy. The computational effort can be reduced by a factor of 4~30, depending on the size of the problem (with increasing the inline nodes the required k_y values remain unchanged as long as the wavenumber range can be covered by the logarithmic scale).

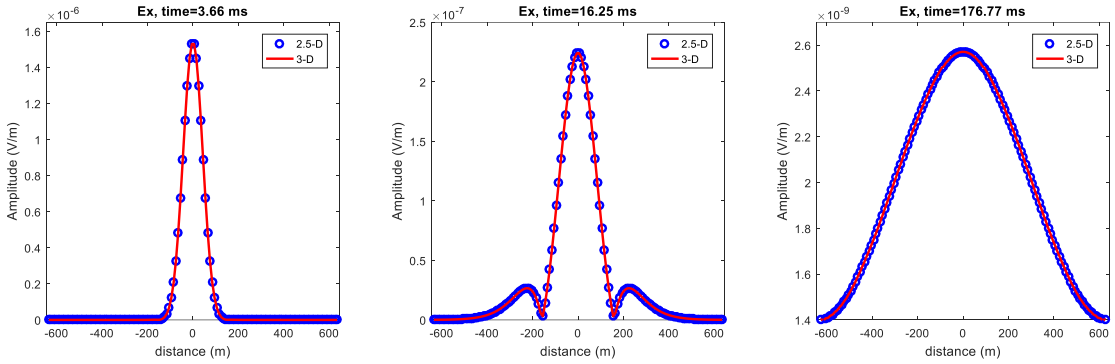

Figure 2 *Comparison of absolute amplitude between the 2.5-D and 3-D modelling results along the receiver line.*

Conclusion

We have extended the theory of REM to solve the 3-D time-domain electric field problem with a dipole source in 1-D/2-D/3-D model space. The example run shows that the computational effort can be reduced by an order of magnitude for a 2.5-D problem without loss of accuracy. We test different approaches to parallelize the code. The GPU-accelerated implementation improves the performance by a factor of over 100, which demonstrates a successful application of GPU computing. The significant improvement in numerical efficiency generalizes the use of REM in large-scale, time-domain EM modelling or inversion problems.

References

Carcione, J. M., 2006, Geophysical software and algorithms: A spectral numerical method for electromagnetic diffusion: *Geophysics*, **71**, no 1, 11-19.

Commer, M., and G. Newman, 2004, A parallel finite-difference approach for 3D transient electromagnetic modelling with galvanic sources: *Geophysics*, **69**, no. 5, 1192-1202.

Frigo, M., and S. G. Johnson, 2005, The Design and Implementation of FFTW3: *Proceedings of the IEEE*, **93**, no. 2, 216-231.

Key, K., and J. Ovall, 2011, A parallel goal-oriented adaptive finite element method for 2.5-D electromagnetic modelling: *Geophys. J. Int.*, **186**, no. 1, 137-154.

Liu, Y., and A. Ziolkowski, 2018, Time evolution of the electric field using the rapid expansion method: inclusion of the free surface and anisotropy: 80th EAGE Conference and Exhibition, EAGE, Expanded Abstract.

Pestana, R. C., and P. L. Stoffa, 2010, Time evolution of the wave equation using rapid expansion method: *Geophysics*, **75**, no. 4, T121-T131.

Stoffa, P. L., and A. Ziolkowski, 2018, Time evolution of the electric field using the rapid expansion method (REM) with pseudo-spectral evaluation of spatial derivatives: 80th EAGE Conference and Exhibition, EAGE, Expanded Abstract.

Um, E. S., J. M. Harris, and D. L. Alumbaugh, 2010, 3D time-domain simulation of electromagnetic diffusion phenomena: A finite-element electric-field approach: *Geophysics*, **75**, no. 4, F115-F126.